



IN THE UNITED STATES

PATENT AND TRADEMARK OFFICE

RECEIVED

AUG 26 2004

Technology Center 2600

APPLICANT: Reuel W. Nash

APPLICATION NO.: 10/074,150

FILING DATE: February 11, 2002

TITLE: MANAGEMENT OF LIMITED RESOURCES IN A GRAPHICS SYSTEM

EXAMINER: Kee M. Tung

GROUP ART UNIT: 2676

ATTY. DKT. NO.: 22278-05791

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below:

Dated: August 16, 2004

By: 

Robert A. Hulse, Reg. No. 48,473

MAIL STOP AMENDMENT
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450

RESPONSE B

Sir:

In response to the first Office Action, mailed May 14, 2004, Applicant submits the following remarks. On the basis of the following remarks, consideration of this application and allowance of all pending claims are requested.

Claims 1-38 were presented for examination and were pending in this application. In the latest Office Action, claims 1, 2, 6, 9-11, 13, 18, 20, 23, 26, 29, 32, 34, 35, and 38 were rejected as being obvious over U.S. Patent No. 5,969,728 to Dye et al. in view of U.S. Patent No.

6,339,427 to Laksono et al., in further view of U.S. Patent No. 6,157,996 to Christie et al. Claims 3-5, 7, 8, 14-17, 19, 21, 22, 24, 25, 27, 30, 31, 33, 36, and 37 were rejected as being obvious over Dye in view of Laksono, in further view of Christie, and in further view of U.S. Patent No. 6,587,113 to Baldwin et al. Claim 12 was rejected as being obvious over Dye in view of Laksono, in further view of Christie, and in further view of U.S. Patent No. 5,315,696 to Case et.al. With respect to claim 28, however, the Office Action provides no reasons for rejecting that claim.

I. Draw Commands in List associated with Predicate Functions or other Conditions

Various embodiments of the claimed invention include systems, products, and methods that generate draw commands to be rendered by a graphics system to create a frame. Rather than waiting until the graphics system is ready to process the draw commands before generating them, the draw commands can be added to a draw command list at an earlier time. To control the timing of the draw command processing, entries in the draw command list may be associated with one or more predicate functions or may otherwise be made conditional upon the occurrence of a particular event or events. In this way, the graphics system can be prevented from attempting to render a draw command before the system is ready – e.g., before a frame buffer can be written to or before a needed texture has been loaded. When the predicate functions or other conditions are satisfied for a particular entry in the list, the associated draw commands are processed (e.g., rendered) by the graphics system. The graphics application can thus “run ahead” by filling the draw command list with draw commands, using buffering and synchronization techniques to reduce the processing time that would otherwise be wasted while waiting for events to occur (such as a swap event and texture loading).

Claim 1, for example, recites a method for synchronizing graphics commands in a multi-stage graphics system that includes the steps, “associating one or more predicate functions with at least some entries in the [draw command] list, each predicate function satisfiable upon the occurrence of a condition,” and, “responsive to the satisfaction of the predicate functions associated with each entry, transferring the draw commands in the entry to a next stage in the graphics system.” Similarly, claim 18 recites, “queuing the draw commands for being transferred to graphics hardware, wherein at least some of the draw commands are associated with one or more conditions,” and, “transferring each draw command to a next stage of the graphics system upon the satisfaction of all of the draw command’s conditions.” Claim 23 recites a computer program product that includes “a draw entry module adapted to maintain a draw command list of draw entries, each draw entry containing one or more draw commands, wherein at least some draw entries further contain one or more predicate functions satisfiable upon the occurrence of a condition” and “a draw entry logic module adapted to transfer each draw entry’s draw commands for processing if all of the draw entry’s predicate functions are satisfied.” Finally, claims 29 and 35 recite a computer program development environment and an associated method, respectively, for producing such a computer program product.

In rejecting these claims, the examiner proposed a combination of Dye’s graphics system with Laksono’s display command list, and further with Christie’s disclosure of predicated execution. Dye describes a graphics system that renders draw commands from “display lists” into frame buffers, which contain the pixel values for a frame shown on a computer display. Laksono describes a specific method of managing a display command list, similar to Dye’s “display lists,” which contains a set of graphics commands to be processed in a graphics system. But as the examiner acknowledged in the Office Action, Neither Dye nor Laksono discloses or

suggests associating draw commanded in the list with a predicate function or other conditions that must be satisfied before the particular draw command or commands can be processed. To cure this deficiency in the proposed combination, the examiner cited Christie. But Christie merely describes a general purpose processor that either executes an instruction or skips it based on the value of a “predication update bit field.” As shown below, this combination of references fails to suggest each of the claimed limitations, and the combination itself is improper.¹

A. The proposed combination of references does not suggest the claimed invention

The specification explains that the claimed predicate functions “specify conditions that must be satisfied before an associated draw command is transferred to the next stage in the graphics system.” (See the specification, for example, at ¶ 12.) In this way, draw commands are not processed until after particular conditions have been satisfied. Examples of such conditions include the loading of a texture required for a draw command, or the occurrence of a swap event that frees a frame buffer so that a draw command can be rendered to write to the pixels of that frame buffer.

Christie’s disclosure of “predicated execution” is not the same as nor does it suggest the claimed conditional processing of draw commands. Christie discloses a microprocessor that either executes or skips instructions based on the value of a “predication update bit field.” As Christie explains:

Predicated execution of a computer instruction implies that, prior to actually executing the pending computer instruction, an execution unit of the microprocessor checks a particular condition within the microprocessor to determine if the pending instruction is to be executed. If the condition represented by the predicated execution information in the decoded instruction is false, the execution unit completes the instruction without executing it.

¹ Although the rejections of some of the claims relied upon additional references, such as Baldwin and Case, these additional references were cited for their disclosure of particular dependent limitations, and not the limitations argued in this section. As such, those additional references are not relevant to the arguments in this section, which pertain to the limitations of the independent claims.

(Christie, col. 8, line 66 to col. 9, line 6.) Thus, Christie's microprocessor executes an instruction when an associated flag (the "predication update bit field") is set, and it skips execution of the instruction when the flag is not set. Accordingly, Christie is not concerned with *when* to execute a command, but rather *whether* to execute it. At best, therefore, implementing Christie's "predicated execution" function onto Dye and Laksono would result in a system that is able to skip the execution of certain draw commands. But this contrasts sharply with the claimed invention, which is directed at causing the graphics system to wait until an appropriate time before processing a draw command. Importantly, the proposed combination of Christie with Dye and Lanksono still fails to achieve the claimed conditional processing function.

This fundamental difference becomes even clearer by observing that Christie's "predication update bit field" (also referred to as the "predicated update field") is just a bit field in the machine specific register (MSR) of the processor. (Christie, col. 10, lines 44-61.) In the embodiment described in Christie, this bit field consists of a single bit that specifies whether the PFLAG register should be set upon execution of a machine instruction. Unlike the claimed predicate functions, the single bit of Christie's predication update bit field is a data storage element. Since Christie's predication update bit field simply contains true/false information, it is not capable of being a predicate function that is satisfiable upon the occurrence of a condition or conditions. Therefore, even if one were to combine Christie's predication update bit field with Dye's graphics system and Laksono's draw command list entries, as proposed, this combination would fail to achieve the claimed association of conditions that are satisfied before draw commands are processed.

Accordingly, claims 1-38 are patentable over the proposed combinations.

B. There would have been no motivation to combine Christie as proposed

In addition, the proposed combination of Dye, Laksono, and Christie is improper because there would have been no reason for a person of ordinary skill in the art at the time the invention was made to combine the references as the examiner suggested. Merely because two references *can* be combined or that the combination is feasible within the skill in the art does not render the resulting combination obvious. “Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching, suggestion or incentive supporting the combination.” *In re Geiger*, 815 F.2d at 688 (Fed. Cir. 1987); *see also* MPEP §§ 2141, 2143.01. Here, there is no such motivation.

In particular, there is no motivation to combine the use of Christie’s predication update bit field with draw command entries in a draw command list. It is first noted that the suggestion to turn to predicate functions is completely absent from both Dye and Laksono. Conversely, Christie fails to suggest applying its “predicated execution” techniques to entries in a draw command list. Indeed, Christie makes no mention of draw commands, as the subject matter of Christie does not relate to graphics. More specifically, there is no suggestion in Christie of applying the predication update field to list entries in general, nor is there any suggestion of using the predication update field to control the transfer of a data entry, of any type, to the next stage of a system.

The only justification given by the examiner for making this combination was “to avoid the performance penalty in a multi-stage pipelined processor,” citing Christie at col. 2, lines 23-56. But this justification is insufficient. In this passage, Christie is actually talking about conditional execution of instructions, not predicate functions that must be satisfied before draw commands can be processed. By avoiding “the performance penalty,” Christie is referring to the

penalty the results when instructions already loaded in a pipeline are rendered unnecessary because the code branches and no longer needs to execute those instructions. This purpose is fundamentally different from how the examiner has sought to use Christie – i.e., as a predicate for draw commands to control the timing of the processing of those draw commands. The only common benefit is some vague notion of improving performance, but this insufficient to demonstrate the required motivation. To make the combination, there must exist a more specific reason to combine the references in the manner proposed in the Office Action. The passage of Christie upon which the examiner relies, like the rest of Christie, lacks such a motivation.

Because the proposed combinations are improper, claims 1-38 are patentable.

II. Processing Draw Commands based on status of Texture Loading

Claims 11, 16, 20, 26, and 32 recite limitations related to processing a draw command based on the status of the loading of a texture. For example, claim 11 recites “associating a predicate function with the entry in the draw command list that holds the draw command associated with the texture, the predicate function satisfied responsive to the texture’s being loaded into a texture memory of the graphics system.” In this way, a draw command can be processed before a texture that it needs is loaded, and the system can ensure that the draw command will not be processed until the texture is loaded.

Processing a draw command based on whether a texture has been loaded is not disclosed or suggested in any of the references cited in the Office Action. In rejecting dependent claims 11, 16, 20, 26, and 32, the examiner cited Dye at col. 5, line 52 for this additional claimed feature. But this passage in Dye simply mentions that the graphics system memory 118 (shown in FIG. 1 of Dye) may contain texture maps – a far cry from suggesting that a predicate function associated with a draw command is satisfied upon the loading of a texture.

Accordingly, claims 11, 16, 20, 26, and 32 are patentable for the additions reasons set forth herein.

III. Texture Load List

In rejecting claims 4, 25, 31, and 37, the examiner asserted that Baldwin teaches adding instructions for loading a texture into a texture load list and loading textures into the texture memory according to the texture load list when texture memory is available. However, Baldwin does not teach either of these ideas, nor does it even mention a texture load list anywhere in the specification. The cited text in Baldwin (column 21, lines 20-50) discusses cache loading options, not a load list that contains instructions for loading textures.

Accordingly, claims 4, 5 (which depends from claim 4), 25, 31, and 37 are patentable for the additions reasons set forth herein.

IV. Summary

Based on the foregoing, the application is in condition for allowance of all claims, and a Notice of Allowance is respectfully requested. If the examiner believes for any reason direct contact would help advance the prosecution of this case to allowance, the examiner is encouraged to telephone the undersigned at the number given below.

Respectfully submitted,

REUEL W. NASH

Dated: August 16, 2004

By: 

Robert A. Hulse, Reg. No. 48,473
Attorney for Applicant
Fenwick & West LLP
801 California Street
Mountain View, CA 94041
Tel.: (415) 875-2444
Fax: (415) 281-1350